PEGRL: Improving Machine Translation by Post-Editing Guided Reinforcement Learning

Yunzhi Shen[1] **Hao Zhou**[1] **Xin Huang**[2*] **Xue Han**[2] **Junlan Feng**[2]
Shujian Huang[1*]
[1]National Key Laboratory for Novel Software Technology, Nanjing University
[2]China Mobile Research Beijing, China
{shenyunzhi, zhouh}@smail.nju.edu.cn huangsj@nju.edu.cn
{huangxin, hanxuejt, fengjunlan}@cmjt.chinamobile.com

Abstract

Reinforcement learning (RL) has shown strong promise for LLM-based machine translation, with recent methods such as GRPO demonstrating notable gains; nevertheless, translation-oriented RL remains challenged by noisy learning signals arising from Monte Carlo return estimation, as well as a large trajectory space that favors global exploration over fine-grained local optimization. We introduce **PEGRL**, a *two-stage* RL framework that uses post-editing as an auxiliary task to stabilize training and guide overall optimization. At each iteration, translation outputs are sampled to construct post-editing inputs, allowing return estimation in the post-editing stage to benefit from conditioning on the current translation behavior, while jointly supporting both global exploration and fine-grained local optimization. A task-specific weighting scheme further balances the contributions of translation and post-editing objectives, yielding a biased yet more sample-efficient estimator. Experiments on English→Finnish, English→Turkish, and English↔Chinese show consistent gains over RL baselines, and for English→Turkish, performance on COMET-KIWI is comparable to advanced LLM-based systems (DeepSeek-V3.2). Our code and a set of representative pretrained models are publicly available at https://github.com/NJUNLP/peg-rl and https://huggingface.co/collections/DGME/pegrl.

1 Introduction

Reinforcement learning (RL) techniques on large language models (LLMs) have achieved notable advances, exemplified by DeepSeek-R1 (DeepSeek-AI et al., 2025a), which demonstrates strong performance on verifiable tasks such as mathematical reasoning and code generation. More recently, RL-based methods, such as GRPO (Shao et al., 2024), have been adapted for machine translation through the use of automatic evaluation metrics, including BLEU (Post, 2018) and COMET-style metrics (Rei et al., 2022, 2023), as reward signals (He et al., 2025; Feng et al., 2025). Despite these initial improvements, Zeng et al. (2025) show that the Monte Carlo group-wise baseline used in GRPO may suffer from high estimation variance, causing instability in training and suggesting opportunities for further refinement.

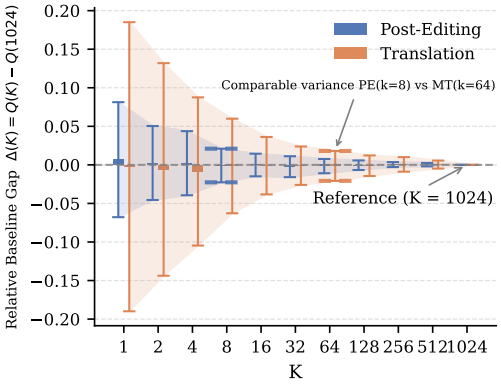

Figure 1: Convergence of the GRPO *group-wise baseline* with respect to the number of sampled trajectories K. For each of 100 instances, we roll out 1024 trajectories and use the resulting baseline as a reference. We report the mean and standard deviation (error bars) of the relative gap $\Delta(K) = Q(K) - Q(1024)$, where K denotes the GRPO group size. Larger K reduces Monte Carlo variance (Appendix B.1), making $Q(1024)$ a potential proxy for the true baseline $\mathbb{E}[R]$. Smaller error bars indicate more stable baseline estimation.

Moreover, the large trajectory space in translation-oriented RL tends to emphasize **global exploration**, while providing limited optimization signals for fine-grained local improvements. Thus the corresponding translation quality is limited, especially for those low-resource translation directions, or those models that are not thoroughly trained.

Compared to machine translation, post-editing

*Co-corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 17225–17242

refines an existing target-side draft with typically minor edits (Melby, 1984; Do Carmo et al., 2021; Lim et al., 2025), enabling **exploration within a more localized output neighborhood** for a given translation trajectory. As shown in Figure 1, post-editing also exhibits substantially lower baseline variance than translation, indicating potentially smaller policy gradient variance and more stable training.

We propose to model the translation workflow as a two-step process: translation followed by post-editing. This allows post-editing to perform fine-grained exploration of the output space based on the initial translation trajectory for improved translations. As a subsequent stage, the post-edited outputs directly reflect the quality of the edited translation, providing more stable learning signals for optimizing the translation policy, which helps mitigate the noise introduced by return estimation in the translation task itself.

This workflow is formulated as a two-stage RL problem. Under Monte Carlo sampling, the joint policy gradient decomposes into additive contributions from translation and post-editing, naturally aligning with the intuition outlined in the previous paragraph (see Section 3 for details). Motivated by variance considerations in return estimation, we introduce a task-specific weighting scheme that places greater emphasis on the post-editing learning signal, whose baseline provides a more stable estimate of the optimized return, while down-weighting the translation term that involves additional variability. Although this results in a biased estimator, we demonstrate both theoretically and empirically that it is more sample-efficient than its unbiased counterpart. To optimize the weighted objective, we introduce **PEGRL**, a GRPO-based dual-task training framework in which translation produces on-policy data for post-editing at each iteration. This design enables comprehensive exploration while ensuring that the post-editing objective, whose return estimation benefits from conditioning on the current translation policy, is optimized under up-to-date translation behavior. Our experiments further show that local exploration induced by post-editing promotes more efficient global exploration (see Section 6.1).

We evaluate our approach on English→Finnish, English→Turkish, and English↔Chinese translation using the WMT24 and FLORES benchmarks. Across chrF++, COMETKIWI, and XCOMET, our method consistently outperforms the RL baseline MT-R1-Zero (Feng et al., 2025), with particularly strong gains in less-covered language directions for the base model (EN→FI and EN→TR). Notably, on English→Turkish, our COMET-KIWI scores are competitive with state-of-the-art LLMs such as DeepSeek-V3.2 (DeepSeek-AI et al., 2025b). These results demonstrate the effectiveness of our framework in leveraging more stable learning signals to improve translation quality. Our main contributions are as follows:

- We analyze the policy gradients of post-editing and show that, under GRPO, the corresponding baseline is substantially easier to estimate than that of direct translation.
- We propose a two-stage translation framework that integrates translation and post-editing to enable joint global and local RL exploration, with task-specific gradient weighting that exploits the lower-variance post-editing signal for more stable and sample-efficient learning.
- We implement a GRPO-based dual-task RL framework and demonstrate its effectiveness on WMT24 and FLORES datasets (EN→FI, EN→TR, EN↔ZH), outperforming strong RL baselines, and achieving performance on some metrics and directions comparable to SOTA LLMs.

2 Related Work

LLMs for Post-Editing LLMs have shown strong inference-time post-editing performance on WMT benchmarks (Raunak et al., 2023), but training-time LLM post-editing remains underexplored. *Mufu* (Lim et al., 2025) uses a teacher–student setup with auxiliary translations but relies on a strong teacher and surface metrics. In contrast, we model post-editing as a learned policy within a unified RL framework, evaluated with both lexical and semantic metrics.

RL for Machine Translation Inspired by RL successes on verifiable reasoning tasks (DeepSeek-AI et al., 2025a), recent work adapts RL to translation using GRPO-style optimization with diverse reward designs. For example, R1-T1 (He et al., 2025) combines COMET-based rewards with format signals, MT-R1-Zero (Feng et al., 2025) uses hybrid BLEU+COMET rewards, and DeepTrans (Wang et al., 2025) and SSR-Zero (Yang et al., 2025b) adopt trajectory-level generative rewards. These works focus primarily on reward design, while tra-

jectory sampling and multi-stage or multi-task setups, which can significantly affect translation performance, have received less attention.

RL Algorithms for LLMs Policy gradient methods for LLM post-training optimize expected reward:

$$\mathcal{J}_\mu(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\cdot|q)}[R(\tau \mid q)],$$
$$\nabla_\theta \mathcal{J}_\mu(\theta) = \mathbb{E}_\tau[\widehat{A}(\tau, q)\, \nabla_\theta \log \pi_\theta(\tau \mid q)],$$

with different methods computing the advantage $\widehat{A}$. PPO (Schulman et al., 2017) uses GAE (Schulman et al., 2018), while GRPO (Shao et al., 2024) normalizes rewards over a group.

3 Formal Framework

We formulate machine translation and post-editing as sequential decision processes within a unified RL framework. Let q denote the initial translation prompt, and let $\tau_0 = (a_0, a_1, \ldots, a_{|\tau_0|})$ be the translation trajectory, where each a_i is a translation token. Conditioned on τ_0, the model generates a post-editing trajectory $\tau_1 = (b_0, b_1, \ldots, b_{|\tau_1|})$, where each b_i is a post-editing token. The post-editing policy is additionally conditioned on an auxiliary prompt p, which, together with q, is derived from the same source input.

Let π_θ denote the LLM with parameters θ. We optimize a trajectory-level RL objective:

$$\max_\theta \; \mathbb{E}_{\tau_0\sim\pi_\theta(\cdot|q),\, \tau_1\sim\pi_\theta(\cdot|p,\tau_0)}\big[R(\tau_1)\big]. \quad (1)$$

where the reward $R(\tau_1)$ is assigned to the post-editing trajectory. The policy gradient of this objective is given by (see Appendix A for details):

$$\begin{aligned}&\nabla_\theta \mathbb{E}_{\tau_0\sim\pi_\theta(\cdot|q),\, \tau_1\sim\pi_\theta(\cdot|p,\tau_0)}\big[R(\tau_1)\big] \\ &= \mathbb{E}_{\tau_0,\tau_1}\big[\nabla_\theta \log \pi_\theta(\tau_1 \mid p, \tau_0)\, R(\tau_1)\big] \\ &\quad + \mathbb{E}_{\tau_0}\big[\nabla_\theta \log \pi_\theta(\tau_0 \mid q)\; \mathbb{E}_{\tau_1}[R(\tau_1)]\big]. \quad (2)\end{aligned}$$

3.1 Two-stage Monte-Carlo Estimation

The policy gradient in the right hand side of Eq. (2) involves nested expectations over τ_0 and τ_1, which are intractable to compute exactly. To address this, we adopt a two-stage Monte Carlo estimator (Metropolis et al., 1953) that removes the double expectation.

Given a query q, we first sample N trajectories $\{\tau_0^{(i)}\}_{i=1}^N$ from $\pi(\cdot \mid q)$. For each $\tau_0^{(i)}$, we then sample M trajectories $\{\tau_1^{(i,j)}\}_{j=1}^M$ from $\pi(\cdot \mid p, \tau_0^{(i)})$.

We refer to the following term as the *post-editing policy gradient*. Using Monte Carlo sampling and expanding only the expectation over τ_0, **the inner expectation reduces to a standard policy gradient** for post-editing conditioned on a fixed input $\tau_0^{(i)}$, derived via the log-derivative trick (Appendix A.1).

$$\begin{aligned}&\mathbb{E}_{\tau_0}\Big[\mathbb{E}_{\tau_1}[\nabla_\theta \log \pi_\theta(\tau_1 \mid p, \tau_0)\, R(\tau_1)]\Big] \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\tau_1}[\nabla_\theta \log \pi(\tau_1 \mid p, \tau_0^{(i)})\, R(\tau_1)]\end{aligned}$$

Analogously, we refer to the following term as the *translation policy gradient*. Expanding only the expectation over τ_1 yields **the policy gradient of the translation task** with respect to the input q, derived via the log-derivative trick (Appendix A.1).

$$\begin{aligned}&\mathbb{E}_{\tau_0}\Big[\nabla_\theta \log \pi_\theta(\tau_0 \mid q)\; \mathbb{E}_{\tau_1}[R(\tau_1)]\Big] \\ &\approx \mathbb{E}_{\tau_0^{(i)}}\Big[\nabla_\theta \log \pi_\theta(\tau_0^{(i)} \mid q)\; \frac{1}{M}\sum_{j=1}^{M} R(\tau_1^{(i,j)})\Big].\end{aligned}$$

3.2 Optimization with GRPO

Following the decomposition in Section 3.1, we estimate both policy gradients using GRPO. For post-editing, the group-normalized advantage is computed directly from the post-editing reward $R(\tau_1)$. For translation, we use the average reward of the associated post-editing candidates to compute the group-normalized advantage for updating the translation policy.

$$\bar{R}_{\text{pe}}^{(i)} \;=\; \frac{1}{M}\sum_{j=1}^{M} R\Big(\tau_1^{(i,j)}\Big), \quad (3)$$

where $\tau_1^{(i,j)}$ denotes the j-th post-editing trajectory associated with the i-th translation sample. Formally, this guides Stage 1 toward optimization directions that improve Stage 2 output quality.

3.3 Variance Analysis of RL Baseline

As illustrated in Section 1 and Fig. 1, starting from the baseline construction in GRPO advantage estimation, for a fixed draft trajectory τ_0, the post-editing baseline $\mathbb{E}_{\tau_1\sim\pi_\theta(\cdot|\tau_0,p)}\big[R(\tau_1)\big]$ provides a more accurate estimate than the translation-level baseline $\mathbb{E}_{\tau_0\sim\pi_\theta(\cdot|q)}\big[R(\tau_0)\big]$. Moreover, **the translation gradient discussed**

in Section 3.1 requires estimating the nested expectation $\mathbb{E}_{\tau_0 \sim \pi_\theta(\cdot|q),\, \tau_1 \sim \pi_\theta(\cdot|p,\tau_0)}\big[R(\tau_1)\big]$.

The variance of the estimator, $\mathrm{Var}_{\tau_0 \sim \pi_\theta(\cdot|q),\, \tau_1 \sim \pi_\theta(\cdot|\tau_0,p)}[R(\tau_1)]$, decomposes into a non-negative between-τ_0 term and $\mathbb{E}_{\tau_0}[\mathrm{Var}_{\tau_1|\tau_0}(R(\tau_1))]$ (Appendix B). The latter corresponds exactly to the variance of the post-editing estimator conditioned on a fixed τ_0, i.e., $\mathrm{Var}_{\tau_1 \sim \pi_\theta(\cdot|\tau_0,p)}[R(\tau_1)]$. Therefore, conditioning on τ_0 removes the between-τ_0 variability and yields a lower-variance estimator in most cases. Accordingly, within our framework, the post-editing policy gradient baseline provides a lower-variance estimate than the translation policy gradient baseline.

4 Methodology

Based on the theoretical derivations presented earlier, we propose a GRPO-based RL training framework that jointly integrates the training of translation and post-editing. Unlike simple mixed RL training schemes (DeepSeek-AI et al., 2025b), the two tasks in our framework are tightly coupled: the translation component generates training data online for post-editing, while feedback from post-editing guides the translation model toward outputs that better facilitate downstream post-editing. We train a single model with both tasks simultaneously. **In a single training step**, trajectories are sampled from both tasks and contribute carefully weighted gradients (see Section 4.3) for model updates.

4.1 Hybrid Sampling for Online Post-Editing Data Generation

In our framework, translation and post-editing use separate prompts, reflecting the dual-task setup and avoiding the performance drop from multi-task prompts (Khot et al., 2023). The post-editing prompt is conditioned on the translation output and generated online during training (Appendix D).

Thus we perform a hybrid sampling for both tasks. At each training step, for a translation pair (src, tgt), following the sampling procedure in Section 3.1, we obtain N translation trajectories $\{pred_i\}_{i=1}^{N}$ and $N \times M$ post-editing trajectories $\{pe_{i,j}\}_{i=1,j=1}^{N,M}$. In our main experiments, we set $N = M = 8$.

4.2 Reward and Advantage

Our reward function consists of three components. First, the post-editing policy is trained with a quality estimation reward. Second, the translation policy is optimized using the expected reward $\frac{1}{M}\sum_{j=1}^{M} R(\tau_1^{(i,j)})$ from the post-editing task. Finally, we introduce a penalty term to discourage degenerate behaviors, such as unbounded or excessively long outputs.

4.2.1 Reward for Post-editing

The post-editing objective is defined to encourage quality improvements as measured by a quality estimation function $f(\cdot)$. Under the group-relative policy optimization (GRPO) framework, optimizing improvement-based rewards is equivalent to directly optimizing absolute output quality after group-advantage normalization. A formal proof is provided in Appendix C.

To prevent degenerate updates, if the post-edited output does not modify the initial translation ($pe_{i,j} = pred_i$) and its estimated semantic quality falls below a threshold α (e.g., $\alpha = 0.95$, which is used in all our experiments), we assign a zero reward. Let $\mathcal{D}(u)$ denote this condition. For each post-editing instance $u = (src, pred_i, pe_{i,j}, tgt)$, the post-editing reward is defined as

$$R_{\mathrm{pe}}(u) = \begin{cases} 0, & \mathcal{D}(u), \\ f(pe_{i,j} \mid src, tgt), & \text{otherwise.} \end{cases} \tag{4}$$

In our subsequent experiments, $f(\cdot)$ is instantiated by COMETKiwi (Rei et al., 2023) together with a surface-level metric, e.g., chrF++ (Popović, 2017) or BLEU(Post, 2018).

4.2.2 Reward For Translation

When computing the translation reward, for each translation instance $v = (src, pred_i, tgt)$, we aggregate the contributions from all associated post-editing trajectories. Let $\mathcal{C}(v)$ denote the set of post-editing trajectories corresponding to v. The translation reward is then defined as

$$R_{\mathrm{mt}}(v) = \mathrm{Mean}\big(\{R_{\mathrm{pe}}(u) \mid u \in \mathcal{C}(v)\}\big). \tag{5}$$

This formulation directly corresponds to the average post-editing reward defined in Eq. (3).

4.2.3 Penalty Reward

We disable explicit reasoning in Qwen3 (Yang et al., 2025a), and thus do not use CoT during trajectory generation (Wei et al., 2023). To discourage degenerate behaviors such as excessive repetition or unbounded outputs, any such trajectory is assigned a total reward of -1.

MODEL	EN–FI (WMT24)			EN–FI (FLORES200)			EN–TR (WMT24)			EN–TR (FLORES200)		
	chrF++	Kiwi	xCOM	chrF++	Kiwi	xCOM	chrF++	Kiwi	xCOM	chrF++	Kiwi	xCOM
Resource-Constrained LLM-based Translation Systems												
General-purpose LLMs												
Qwen3-4B	40.74	41.27	45.86	36.79	46.87	48.77	40.12	50.60	53.89	42.34	61.61	65.91
Qwen3-8B	45.86	51.92	58.15	43.28	61.77	66.72	44.82	59.25	63.04	47.58	70.62	76.89
Qwen3-14B	49.02	60.43	66.80	46.48	70.06	77.34	47.56	63.26	67.82	50.25	73.98	82.39
Qwen3-32B	48.69	60.54	67.34	46.61	71.10	78.55	47.18	62.66	66.47	49.46	73.28	81.32
MT-R1-Zero												
MT-R1-Zero-4B	43.42	56.04	61.34	40.49	65.20	69.41	43.25	61.57	63.85	45.22	72.70	78.14
MT-R1-Zero-8B	47.45	62.16	69.79	44.51	72.42	78.78	47.10	63.96	68.98	48.72	75.92	83.53
Ours												
Ours-4B	45.29	62.49	69.40	42.65	73.78	79.99	45.39	65.49	69.24	47.77	76.35	83.63
Ours-8B	**49.02**	**67.90**	**76.49**	**46.62**	**79.07**	**86.50**	**48.04**	**68.14**	**73.51**	**50.41**	**78.26**	**87.25**
LLM-based Translation Systems with Large Models or Extensive Data (only for reference)												
General-purpose LLMs												
Gemini-2.0-flash	57.93	75.74	87.09	58.09	85.72	95.83	**57.42**	68.05	77.65	59.46	79.48	92.10
OpenAI GPT-5.2	**59.44**	**76.26**	**87.83**	59.56	**86.53**	**96.01**	56.14	**69.45**	**77.87**	58.68	**79.96**	**92.39**
DeepSeek-V3.2	57.18	74.00	85.87	56.53	84.71	94.70	56.21	68.13	77.84	58.38	79.55	91.70
Translation-specific LLMs												
Seed-X-PPO-7B	57.48	74.72	86.51	**62.57**	85.53	95.32	54.28	67.28	75.99	**62.76**	78.94	91.40
TowerInstruct-13B-v0.1	44.58	53.74	61.81	43.96	68.79	76.29	-	-	-	-	-	-

Table 1: Results on translation directions (EN–FI and EN–TR). In the metric columns, **xCOM denotes xCOMET**. Models are grouped into resource-constrained LLM-based systems and large-scale or data-intensive LLM-based translation systems. A dash ("–") indicates that the model does not support the corresponding language direction. MT-R1-Zero serves as the baseline, and both *Ours* and MT-R1-Zero are trained with the same amount of data. The best settings within each category are highlighted in **bold**.

4.2.4 Overall Reward and Advantage Computation

Let x denote either a translation or a post-editing instance in our hybrid sampling step. Trajectories exceeding the token budget are penalized with -1. Valid trajectories receive task-specific rewards:

$$R(x) = \begin{cases} -1, & \text{if } x \text{ exceeds token budget,} \\ R_{\text{pe}}(x), & \text{if } x = (src, pred_i, pe_{i,j}, tgt), \\ R_{\text{mt}}(x), & \text{if } x = (src, pred_i, tgt). \end{cases}$$

After reward computation, the translation trajectories $\{pred_i\}_{i=1}^{N}$ form a single GRPO group for advantage computation. The post-editing trajectories $\{pe_{i,j}\}_{i=1,j=1}^{N,M}$ are divided into N GRPO groups, each consisting of $\{pe_{i,j}\}_{j=1}^{M}$ with independently computed advantages. All advantages are then used to optimize the policy via the GRPO policy gradient.

4.3 Variance-Aware Gradient Weighting

As discussed in Section 3.3, conditioning on a fixed draft trajectory τ_0 yields a lower-variance estimator of the expected post-editing return, compared to a translation-level baseline that marginalizes over τ_0. As a consequence, the post-editing term in the policy gradient is associated with a more stable learning signal, while the translation-level term involves additional variability due to uncertainty over τ_0.

Motivated by this discrepancy in the variance of their underlying return estimates and the different roles played by the two gradient terms, we introduce weighting coefficients to explicitly balance their relative contributions in Eq. (2). This leads to a biased estimator, but allows for improved stability during optimization:

$$\mathbb{E}_{\tau_0}\Big[\lambda_{\text{pe}}\,\mathbb{E}_{\tau_1}\big[\nabla_\theta \log \pi_\theta(\tau_1 \mid p, \tau_0)\, R(\tau_1)\big]\Big] + \lambda_{\text{mt}}\,\mathbb{E}_{\tau_0}\big[\nabla_\theta \log \pi_\theta(\tau_0 \mid q)\,\mathbb{E}_{\tau_1}[R(\tau_1)]\big]. \quad (6)$$

In our main experiments, we set $\lambda_{\text{pe}} = M$ and $\lambda_{\text{mt}} = 1$, placing greater emphasis on the post-editing signal, whose baseline is more directly aligned with the optimized return. The effects of different λ_{pe} and λ_{mt} settings are further analyzed in Section 6.2.

5 Experiments

5.1 Experimental Setup

Datasets. Following the capabilities of the base models and their relative coverage over different language pairs, we conduct experiments on two categories of translation directions:

- **Less-Covered Directions.** We conduct experiments with Qwen3-(4B, 8B) (Yang et al.,

2025a) on English→Finnish (EN→FI) and English→Turkish (EN→TR). For EN→FI, 7K sentence pairs are sampled from the validation and test sets of WMT17–19 (Bojar et al., 2017, 2018; Foundation), while for EN→TR, 6K sentence pairs are sampled from the WMT17–18 test sets. For these language directions, the function $f(\cdot)$ in Eq. (4) is defined as the sum of COMETKiwi and chrF++.

- **More-Covered Directions.** We conduct experiments with the smaller Qwen3-0.6B (Yang et al., 2025a) on English↔Chinese (EN↔ZH), where the base model exhibits substantially stronger prior competence. The bidirectional parallel data are collected following prior work (Feng et al., 2025). For these language directions, the function $f(\cdot)$ in Eq. (4) is defined as the sum of COMETKIWI and BLEU.

Across all language pairs, evaluation is conducted on the WMT24 test sets (Deutsch et al., 2025) and the FLORES-200 benchmark (Costa-jussà et al., 2022). In addition, for EN↔ZH, we further report results on a more challenging challenge set collected in prior work (Cheng et al., 2025).

Baselines. Our baselines are grouped into two categories. One category comprises advanced LLM-based translation systems characterized by large model sizes ($\geq$100B parameters) and/or extensive training data, including general-purpose LLMs such as Gemini-2.0-Flash,[1] OpenAI GPT-5.2,[2] and DeepSeek-V3.2 (DeepSeek-AI et al., 2025b), as well as translation-specialized models Seed-X-PPO-7B (Cheng et al., 2025) and TowerInstruct-13B-v0.1 (Alves et al., 2024). The other category targets resource-constrained settings and includes the Qwen3 family of general-purpose models and our primary comparison method, MT-R1-Zero. Unlike our hybrid trajectory design that interleaves translation and post-editing, MT-R1-Zero samples trajectories only at the translation stage. To control variables, we use the same prompts as in MT-R1-Zero and compute its translation quality using the post-editing reward (R_{pe}), reporting results under the **non-thinking** setting.

Evaluation Metrics. We evaluate translation quality along both surface-form and semantic dimensions. For surface-level evaluation, we use chrF++ (Popović, 2017) for Finnish and Turkish, which exhibit rich morphological variation, and BLEU (Post, 2018) for English and Chinese, where BLEU is well established. For semantic evaluation, we adopt cost-effective COMET-style models: COMETkiwi (Rei et al., 2023) as a reference-free metric and XCOMET (Guerreiro et al., 2023) as a reference-based metric. Both metrics are used in their XL variants.

Training Details. We adopt VeRL (Sheng et al., 2024) as the RL training framework. During training, the input prompt length is capped at 768 tokens, and the maximum output length is set to 512 tokens. Gradients are computed with an effective batch size of 128 samples per step using gradient accumulation, and the learning rate is set to 5×10^{-7}.

For GRPO sampling, our approach rolls out 8 translation candidates per input and further rolls out 8 post-editing outputs for each translation, resulting in 72 trajectories per data instance. **Accordingly, all compared methods are trained with 72 rollouts per example to ensure a fair comparison.**

Main experiments are conducted on 1×8 NVIDIA A100 GPUs (80GB) and 4×8 NVIDIA H20 GPUs (96GB). Training for a single language direction takes approximately 24 hours, requiring around 400 training steps.

We further extend our experiments to the Ascend platform by training two 4B-scale models on Ascend 910 GPUs. Detailed implementation and experimental settings on Ascend are provided in Section 6.5 and Appendix E.3.

5.2 Main Results

Our method outperforms pure GRPO under resource constraints. As Table 1 shows, Ours-8B surpasses Qwen3-32B on EN→FI, achieving COMET-KIWI gains of +7.36 (WMT24) and +7.97 (FLORES), with even larger improvements on XCOMET: +9.15 (WMT24) and +7.95 (FLORES). For EN→TR, we observe consistent gains of approximately 5–6 points on COMET-KIWI and around 7 points on XCOMET. Ours-4B also outperforms Qwen3-32B on both COMET-KIWI and XCOMET.

Compared to MT-R1-Zero, our approach delivers larger improvements using the same base models. On EN→FI (WMT24), Ours-4B improves XCOMET by +23.54, compared to +15.48 for MT-R1-Zero-4B, while Ours-8B achieves +18.34 versus +11.64 for MT-R1-Zero-8B. On EN↔ZH (Table 2), our method consistently outperforms MT-

[1] https://ai.google.dev/gemini-api/docs/models
[2] https://platform.openai.com/docs/models/gpt-5.2

Model	EN–ZH (WMT24)			EN–ZH (FLORES)			EN–ZH (Challenge)			ZH–EN (WMT24)			ZH–EN (FLORES)			ZH–EN (Challenge)		
	BLEU	Kiwi	xCOM	BLEU	Kiwi	xCOM	BLEU	Kiwi	xCOM	BLEU	Kiwi	xCOM	BLEU	Kiwi	xCOM	BLEU	Kiwi	xCOM
Qwen3-0.6B	26.20	58.57	64.45	30.25	70.54	77.18	21.67	64.33	63.90	15.00	63.87	75.62	19.32	72.85	88.34	15.52	58.83	62.91
MT-R1-Zero	28.23	62.96	67.16	33.24	73.78	79.83	23.00	66.89	65.28	15.97	**66.86**	77.74	19.66	74.91	89.69	16.88	61.56	63.41
Ours	**29.23**	**64.63**	**68.40**	**34.03**	**74.39**	**80.89**	**24.44**	**68.89**	**67.00**	**16.26**	66.69	**78.28**	**20.68**	**75.49**	**90.48**	**17.16**	**62.34**	**64.63**

Table 2: Results on translation directions (EN↔ZH). In the metric columns, **xCOM denotes xCOMET**. Our method consistently outperforms baselines across different language directions and datasets. The best results are highlighted in **bold**.

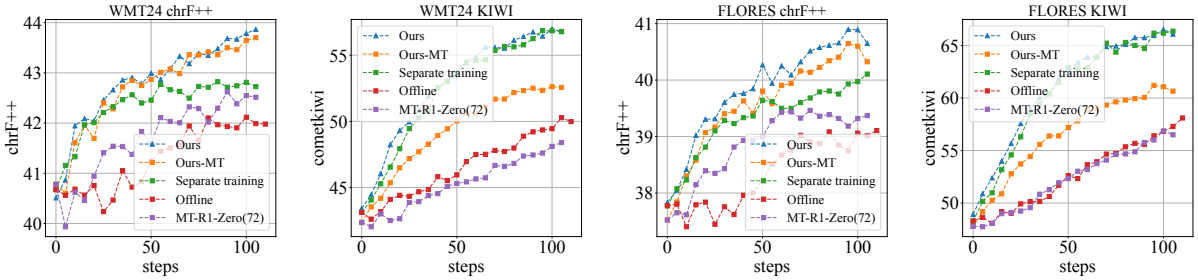

Figure 2: Ablation study of our framework components on WMT24 (EN→FI) and FLORES200 (EN→FI), evaluated using chrF++ and COMET-KIWI. All experiments are conducted on 1K EN→FI translation instances sampled from the training set. In the offline setting, an additional 7K post-editing instances are used. Models are trained for 15 epochs; at each training step, 72 trajectories are sampled per instance, and evaluation is performed every 5 steps.

R1-Zero across most metrics, with only a slight drop on COMET-KIWI for ZH→EN.

Our method achieves strong semantic gains with limited resources. Table 1 shows that Ours-8B approaches state-of-the-art COMET-KIWI performance on EN→TR, closely matching DeepSeek-V3.2 on WMT24 (68.14 vs. 68.13) and FLORES (78.26 vs. 79.55), despite being trained on only 6K examples with an 8B model, demonstrating the effectiveness of our framework.

6 Analysis

6.1 Hybrid Sampling and Reward Analysis

This subsection examines the contribution of each component under different settings.

- **Ours**: Post-editing is trained with online-generated data. The translation trajectories are optimized using rewards derived from post-editing feedback, while the post-editing trajectories are optimized with $R_{\text{pe}}(x)$.
- **Ours-MT**: Trained the same with **Ours**. Evaluation is performed using only the first-stage draft translations, without applying post-editing.
- **Separate training**: Post-editing relies solely on online-generated data. Unlike **Ours**, the translation stage is trained only with the sum of COMETKIWI and chrF++.
- **Offline**: Post-editing is trained on static, pre-collected data, and both translation and post-editing models optimize only sum of COMETKIWI and chrF++.
- **MT-R1-Zero(72)**: Used for comparison with **Ours-MT**, where the number 72 indicates that it uses 72 translation rollouts for gradient updates.

Online generation of post-editing data is effective. As shown in Figure 2, the *Separate training* setting significantly outperforms its offline counterpart on the COMETkiwi metric, and also achieves a marginal improvement on chrF++. This indicates that our framework does **not** simply optimize two independent tasks.

Stage-1 translation reward aligns better with final post-edited quality. Compared with *Separate training*, our method differs only in the Stage 1 reward, defined as $\mathbb{E}_{\tau_1}[R(\tau_1)]$, which accounts for downstream post-editing. This yields an 1-point improvement in chrF++ on the final outputs, while COMETKIWI remains comparable.

Despite a smaller token budget, first-stage drafts from our framework outperform MT-R1-Zero. As shown in Figure 2, *Ours-MT* outperforms *MT-R1-Zero(72)* on chrF++ and COMET-KIWI. Although each sample yields 8 translation and 64 post-editing trajectories, only the 8 drafts contribute to the policy gradient, compared to 72 translation trajectories in MT-R1-Zero. This indi-

		chrF++	KIWI	SUM	Avg_{pe}
Source	“She had a real fear of food waste,” Mr. Coe said.	–	–	–	–
Reference	“Hän todellakin pelkäsi ruoan tuhlaamista,” Coe sanoi.	–	–	–	–
Base (T1)	“Hänellä oli todellinen järkytys ruoan hukkautumisesta,” Coe hakeutui.	0.25	0.4775	0.7321	0.7504
	“She had a genuine shock about the causing of food to drown,” Coe hakeutui(×: to apply, to seek, to make one’s way).				
Base (T2)	“Hänellä oli varsin vakava huuhtola ruoasta,” sanoi herra Coe.	0.23	0.2132	0.4407	0.4697
	“She had a rather serious huuhtola(×: possibly huuhtoutuminen ‘wash-away / leaching’) about food,” said Mr. Coe.				
M-Z (105 s, T1)	"Hänellä oli oltu todellinen huolia ruoan hajoamisesta", herra Coe sanoi.	0.38	0.5243	0.9004	1.0608
	“She had been had real worries about the decomposition of food,” Mr. Coe said.				
M-Z (105 s, T2)	"Hänellä oli todellinen korko ruoan hukkumisesta", herra Coe sanoi.	0.37	0.3219	0.6929	0.9250
	“She had a genuine korko(×: interest rate / heel) about the drowning of food,” Mr. Coe said.				
Ours (105 s, T1)	“Hänellä oli todellinen huoli ruoan häviöstä”, Coe sanoi.	0.40	0.8849	1.2826	1.2287
	“She had a genuine concern about the loss of food,” Coe said.				
↪**post-edit(T1):**	“Hänellä oli todellinen huoli ruoan häviöstä”, Coe sanoi.	0.40	0.8849	1.2826	–
	“She had a genuine concern about the loss of food,” Coe said.				
↪**post-edit(T1):**	“Hänellä oli todellinen huoli ruoan käyttöstä”, Coe sanoi.	0.40	0.4765	0.8806	–
	“She had a genuine concern about food käyttöstä(×: usage / use / utilization),” Coe said.				
Ours (105 s, T2)	“Hänellä oli todellinen huoli ruokaan menettymästä,” Coe sanoi.	0.33	0.5793	0.9083	1.1774
	“She had a genuine concern about ruokaan menettymästä(×: the loss of food),” Coe said.				
↪**post-edit(T2):**	“Hänellä oli todellinen huoli ruoan häviämisestä,” Coe sanoi.	0.36	0.8591	1.2229	–
	“She had a genuine concern about food disappearing,” Coe said.				
↪**post-edit(T2):**	“Hänellä oli todellinen huoli ruoan menetystä,” Coe sanoi.	0.36	0.7472	1.1067	–
	“He had a genuine concern about food menetystä(×: loss / losing),” Coe said.				

Table 3: Case study of model generation behavior. **Base** (T1/T2) denotes two translation trajectories sampled from the base model. **M-Z** (105s, T1/T2) refers to two trajectories produced by MT-R1-Zero after 105 training steps, while **Ours** (105s, T1/T2) are generated by our method. Each trajectory is followed by its **post-editing variants** (↪ post-edit). We analyze one training-set example using MT-R1-Zero and our 105-step checkpoint, selecting two representative trajectories from eight sampled translations. Scores are chrF++ and COMETKIWI (SUM); Avg_{pe} denotes the average over post-edits. English translations are shown beneath each Finnish output. Misspelled Finnish words are left untranslated and annotated as (×: text), where *text* indicates the intended meaning (e.g., *menetystä* (×: loss / losing), denotes a misspelled form of a word meaning “loss” or “losing”.).

cates that post-editing enables fine-grained local exploration that guides translation toward higher-quality regions and indirectly promotes global exploration.

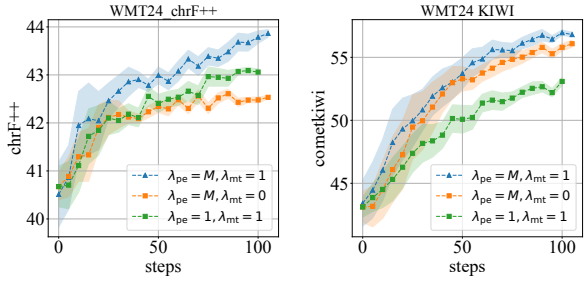

Figure 3: Gradient Weight Analysis. Experimental settings are identical to those in Section 6.1.

6.2 Gradient Weight Analysis

As discussed in Section 4.3, the post-editing and translation gradient terms differ in their noise characteristics due to the variance of their underlying return estimators. In this subsection, we analyze the impact of the scaling factors λ_{pe} (for the post-editing policy gradient) and λ_{mt} (for the translation policy gradient), which control the relative contributions of these two learning signals.

We consider the following experimental settings:

- $\lambda_{\text{pe}} = M$, $\lambda_{\text{mt}} = 1$: Places greater emphasis on the post-editing signal, whose baseline provides a more stable estimate of the optimized return, while keeping the number of trajectories balanced per step.
- $\lambda_{\text{pe}} = 1$, $\lambda_{\text{mt}} = 1$: Treats the two gradient terms equally, yielding an unbiased estimator but with increased sensitivity to noise from the translation-level return estimation.
- $\lambda_{\text{pe}} = M$, $\lambda_{\text{mt}} = 0$: Removes the translation-level term entirely, isolating its contribution to overall performance.

Figure 3 and Table 7 show that $\lambda_{\text{pe}} = M$ and $\lambda_{\text{mt}} = 1$ consistently achieve the best performance on WMT24, yielding the largest gains in chrF++

and improved COMET-KIWI. Accordingly, we adopt this configuration as the default setting in all subsequent experiments.

6.3 LLM-based Evaluation

To assess whether the observed improvements extend beyond standard automatic metrics, we complement our evaluation with an LLM-as-a-judge analysis against MT-R1-Zero. We employ two independent LLM judges, both of which consistently prefer our method on the test set.

We use the following experimental setup:

- *Models.* We compare Ours-8B vs. MT-R1-Zero-8B on en→fi and en→tr, and Ours-0.6B vs. MT-R1-Zero-0.6B on en↔zh.
- *Data & Judges.* Evaluation is conducted on WMT24, using two LLM judges: GPT-5.2 and Gemini-3-Pro-Preview.
- *Prompt Design.* To reduce positional bias, we randomly swap the order of the two candidate translations with equal probability.

As shown in Table 4, LLM preference results consistently favor our method over MT-R1-Zero across all directions and judges, in line with our main findings. While not fully ruling out metric-specific effects, the consistent preferences from two independent LLM judges, together with improvements on X-COMET—an evaluation metric not optimized during training—provide evidence that our method achieves stronger performance than the baseline across diverse evaluation signals, rather than merely better alignment with specific metrics.

Dir.	Method	Gemini	GPT
en→fi	**Ours**	**540/998**	**627/998**
	MT-R1-Zero	310/998	361/998
	T/I	148/998	10/998
en→tr	**Ours**	**570/998**	**592/998**
	MT-R1-Zero	364/998	388/998
	T/I	64/998	18/998
en→zh	**Ours**	**495/998**	**540/998**
	MT-R1-Zero	394/998	449/998
	T/I	109/998	9/998
zh→en	**Ours**	**459/998**	**517/998**
	MT-R1-Zero	453/998	468/998
	T/I	86/998	13/998

Table 4: LLM-based pairwise evaluation. Each entry is reported as a/b, where a denotes the number of LLM-preferred samples and b the total number of evaluated samples; ties or invalid judgments (T/I) may occur.

6.4 Case Study

Base translation explores broadly. Table 3 illustrates two sampled translation trajectories (T1, T2). The base model generates outputs differing in lexical choice and structure, reflecting broad but unstable exploration.

Compared to MT-R1-Zero, our method yields draft translations that are semantically closer to the source and achieves higher average quality after post-editing. Ours (T1/T2) correctly captures the meaning of *food waste* at the draft stage and achieves higher average final output quality (1.2287/1.1774) than MT-R1-Zero (1.0608/0.9250).

6.5 Platform Generalization

To improve the accessibility and reproducibility of our method, we further implement PEGRL on Ascend hardware, with full support for end-to-end training.

We compare the results obtained on the Ascend platform (Atlas 800I A3) at 400 training steps with those from our main experiments, and observe comparable performance, suggesting that our approach generalizes well across different hardware environments. Detailed comparisons are provided in Appendix E.3.

7 Conclusion

We present a two-stage RL framework for machine translation, which models translation and post-editing as sequential actions and enables both global and local RL exploration. By exploiting more stable learning signals derived from conditional return estimation in the post-editing stage, our framework supports more stable policy optimization. Furthermore, a task-specific weighting scheme balances the contributions of translation and post-editing objectives, improving sample efficiency under a fixed token budget. Our results highlight the importance of accounting for variance in return estimation when designing RL objectives, which may be critical for more complex tasks.

8 Limitations

While our framework demonstrates strong performance in translation experiments, its theoretical foundation relies on a task with a relatively small effective sampling space. We have only verified that post-editing can stabilize learning and improve con-

vergence for translation; it remains unclear whether similar auxiliary tasks exist or provide comparable benefits in other domains, such as verifiable-reward tasks, mathematical reasoning, or code generation. Additionally, the reward density of auxiliary tasks in these domains may differ from translation, potentially limiting their impact. In terms of performance, our method still falls short of state-of-the-art LLM-based translation systems, particularly on surface-level metrics, as post-editing often involves minimal changes that are difficult to capture with such metrics. Moreover, due to limited resources, our experiments are restricted to low-resource scenarios and small models; the behavior in high-resource settings remains unexplored.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by National Science Foundation of China (No. 62376116), research project of Nanjing University-China Mobile Joint Institute (NJ20250038), the Fundamental Research Funds for the Central Universities (No. 2024300507), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118).

References

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *Preprint*, arXiv:2402.17733.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ond rej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 272–307, Belgium, Brussels. Association for Computational Linguistics.

Shanbo Cheng, Yu Bao, Qian Cao, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, Wenhao Zhu, Jingwen Chen, Zhichao Huang, Tao Li, Yifu Li, Huiying Lin, Sitong Liu, Ningxin Peng, Shuaijie She, Lu Xu, Nuo Xu, Sen Yang, and 7 others. 2025. Seed-x: Building strong multilingual translation llm with 7b parameters. *Preprint*, arXiv:2507.13618.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025b. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Daniel Deutsch, Eleftheria Briakou, Isaac Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. 2025. WMT24++: Expanding the Language Coverage of WMT24 to 55 Languages & Dialects. *Preprint*, arXiv:2502.12404.

Félix Do Carmo, Dimitar Shterionov, Joss Moorkens, Joachim Wagner, Murhaf Hossari, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2021. A review of the state-of-the-art in automatic post-editing. *Machine Translation*, 35(2):101–143.

Zhaopeng Feng, Shaosheng Cao, Jiahan Ren, Jiayuan Su, Ruizhe Chen, Yan Zhang, Zhe Xu, Yao Hu, Jian Wu, and Zuozhu Liu. 2025. Mt-r1-zero: Advancing llm-based machine translation via r1-zero-like reinforcement learning. *Preprint*, arXiv:2504.10160.

Wikimedia Foundation. Acl 2019 fourth conference on machine translation (wmt19), shared task: Machine translation of news.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Preprint*, arXiv:2310.10482.

Minggui He, Yilun Liu, Shimin Tao, Yuanchang Luo, Hongyong Zeng, Chang Su, Li Zhang, Hongxia Ma, Daimeng Wei, Weibin Meng, Hao Yang, Boxing Chen, and Osamu Yoshie. 2025. R1-t1: Fully incentivizing translation capability in llms via reasoning learning. *Preprint*, arXiv:2502.19735.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. *Preprint*, arXiv:2210.02406.

Zheng Wei Lim, Nitish Gupta, Honglin Yu, and Trevor Cohn. 2025. Mufu: Multilingual fused learning for low-resource translation with llm. *Preprint*, arXiv:2409.13949.

Alan K. Melby. 1984. Machine translation with post editing versus a three-level integrated translator aid system. In *Proceedings of the International Conference on Methodology and Techniques of Machine Translation: Processing from words to language*, Cranfield University, UK.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Vikas Raunak, Amr Sharaf, Yiren Wang, Hany Hassan Awadallah, and Arul Menezes. 2023. Leveraging gpt-4 for automatic translation post-editing. *Preprint*, arXiv:2305.14878.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Nuno M. Guerreiro, José Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André F. T. Martins. 2023. Scaling up cometkiwi: Unbabel-ist 2023 submission for the quality estimation shared task. *Preprint*, arXiv:2309.11925.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-dimensional continuous control using generalized advantage estimation. *Preprint*, arXiv:1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Jiaan Wang, Fandong Meng, and Jie Zhou. 2025. Deeptrans: Deep reasoning translation via reinforcement learning. *Preprint*, arXiv:2504.10187.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Wenjie Yang, Mao Zheng, Mingyang Song, Zheng Li, and Sitong Wang. 2025b. Ssr-zero: Simple self-rewarding reinforcement learning for machine translation. *Preprint*, arXiv:2505.16637.

Guanning Zeng, Zhaoyi Zhou, Daman Arora, and Andrea Zanette. 2025. Shrinking the variance: Shrinkage baselines for reinforcement learning with verifiable rewards. *Preprint*, arXiv:2511.03710.

A Policy Gradient Derivation

We provide the detailed derivation of the policy gradient used in the main text. We first review the log-derivative trick and then apply it to our two-stage trajectory objective.

A.1 Log-Derivative Trick

For a parameterized distribution $p_\theta(x)$ and a scalar function $f(x)$, the gradient of the expectation can

be written as:

$$\begin{aligned}\nabla_\theta \mathbb{E}_{x\sim p_\theta}[f(x)] &= \nabla_\theta \int p_\theta(x)\, f(x)\, dx \\ &= \int \nabla_\theta p_\theta(x)\, f(x)\, dx \\ &= \int p_\theta(x)\, \nabla_\theta \log p_\theta(x)\, f(x)\, dx \\ &= \mathbb{E}_{x\sim p_\theta}\big[\nabla_\theta \log p_\theta(x)\, f(x)\big].\end{aligned}$$

A.2 Two-Stage Trajectory Objective

B Variance Analysis of Monte Carlo Estimators

B.1 Variance of Monte Carlo Estimation

Let $Z \sim P$ and $\mu = \mathbb{E}_{Z\sim P}[f(Z)]$. Given N i.i.d. samples $\{Z_i\}_{i=1}^N$, the Monte Carlo estimator

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} f(Z_i) \tag{7}$$

has variance

$$\mathrm{Var}(\hat{\mu}_N) = \frac{1}{N}\mathrm{Var}_{Z\sim P}[f(Z)]\,. \tag{8}$$

Thus, for fixed N, a larger population variance $\mathrm{Var}[f(Z)]$ results in a higher-variance estimator.

B.2 Law of Total Variance

Let $x \sim p(x)$ and $y \sim q(y \mid x)$. For any function $f(y)$,

$$\mathrm{Var}_{x,y}[f(y)] = \mathbb{E}_x\big[\mathrm{Var}_{y|x}(f(y))\big] \tag{9}$$

$$+\,\mathrm{Var}_x\big(\mathbb{E}_{y|x}[f(y)]\big)\,. \tag{10}$$

The first term captures within-x variability, while the second term reflects variability across different x.

B.3 Variance Ordering of Nested Monte Carlo Estimators

Consider the expectations

$$\mu_0 = \mathbb{E}_{\tau_0\sim\pi_\theta(\cdot|q)}[R(\tau_0)]\,, \tag{11}$$

$$\mu_1(\tau_0) = \mathbb{E}_{\tau_1\sim\pi_\theta(\cdot|\tau_0,p)}[R(\tau_1)]\,, \tag{12}$$

$$\mu = \mathbb{E}_{\tau_0\sim\pi_\theta(\cdot|q),\,\tau_1\sim\pi_\theta(\cdot|\tau_0,p)}[R(\tau_1)]\,. \tag{13}$$

Estimators. Define the Monte Carlo estimators

$$\hat{\mu}_0 = \frac{1}{N}\sum_{i=1}^{N} R(\tau_0^{(i)}), \tag{14}$$

$$\tau_0^{(i)} \sim \pi_\theta(\cdot \mid q), \tag{15}$$

$$\hat{\mu}_1(\tau_0) = \frac{1}{M}\sum_{j=1}^{M} R(\tau_1^{(j)}), \tag{16}$$

$$\tau_1^{(j)} \sim \pi_\theta(\cdot \mid \tau_0, p), \tag{17}$$

$$\hat{\mu} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} R(\tau_1^{(i,j)}), \tag{18}$$

$$\tau_1^{(i,j)} \sim \pi_\theta(\cdot \mid \tau_0^{(i)}, p). \tag{19}$$

Variance comparison. Applying Eq. (10) with $x = \tau_0$ and $y = \tau_1$,

$$\mathrm{Var}_{\tau_0,\tau_1}[R(\tau_1)] = \mathbb{E}_{\tau_0}\big[\mathrm{Var}_{\tau_1|\tau_0}(R(\tau_1))\big] \tag{20}$$

$$+\,\mathrm{Var}_{\tau_0}\big(\mathbb{E}_{\tau_1|\tau_0}[R(\tau_1)]\big)\,. \tag{21}$$

Since the second term is non-negative,

$$\mathrm{Var}_{\tau_0,\tau_1}[R(\tau_1)] \;\geq\; \mathbb{E}_{\tau_0}\big[\mathrm{Var}_{\tau_1|\tau_0}(R(\tau_1))\big]\,. \tag{22}$$

Therefore, under the same sampling budget,

$$\mathrm{Var}(\hat{\mu}) \;\geq\; \mathrm{Var}(\hat{\mu}_1(\tau_0))\,, \tag{23}$$

indicating that conditioning on a fixed τ_0 yields a lower-variance Monte Carlo estimator.

B.4 Other Supporting Evidence

We also empirically approximate that the baseline of post-editing gradients is smaller than that of the MT policy gradients in our framework, as shown in Figure 4.

C Equivalence Between Absolute and Relative Rewards

Theorem 1. *Under GRPO group-advantage normalization, optimizing post-editing rewards defined by absolute quality scores is equivalent to optimizing rewards defined by quality improvements.*

Proof. Let $\mathrm{QE}(\mathrm{pe}_j)$ denote the quality score of the j-th post-editing output, and define the quality improvement $\Delta\mathrm{QE}(\mathrm{pe}_j) = \mathrm{QE}(\mathrm{pe}_j) - C$, where C is a constant baseline shared across all samples in the group.

For a group of M post-editing outputs, the GRPO-normalized advantage is

$$A_j = \frac{\mathrm{QE}(\mathrm{pe}_j) - \mathrm{Mean}(\{\mathrm{QE}(\mathrm{pe}_j)\}_{j=1}^M)}{\mathrm{Std}(\{\mathrm{QE}(\mathrm{pe}_j)\}_{j=1}^M)}.$$

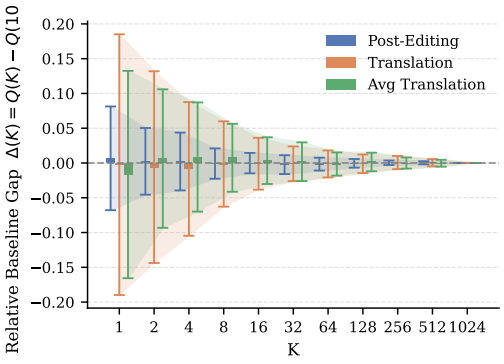

Figure 4: Convergence of the GRPO baseline estimation with respect to the number of sampled trajectories K for post-editing, translation, and average translation (baseline estimation for the translation task in our framework). For each of 100 sampled instances, 1024 trajectories are rolled out and the resulting baseline is used as a reference. The figure reports the mean and standard deviation (error bars) of the relative baseline gap $\Delta(K) = Q(K) - Q(1024)$ computed from the first K trajectories. Smaller error bars indicate lower variance in baseline estimation across instances, corresponding to more stable policy gradient estimates.

Since subtracting a constant does not affect either the mean or the standard deviation within a group, we equivalently obtain

$$A_j = \frac{\Delta\mathrm{QE}(\mathrm{pe}_j) - \mathrm{Mean}(\{\Delta\mathrm{QE}(\mathrm{pe}_j)\}_{j=1}^{M})}{\mathrm{Std}(\{\Delta\mathrm{QE}(\mathrm{pe}_j)\}_{j=1}^{M})}.$$

Therefore, maximizing the GRPO objective based on absolute quality scores is equivalent to maximizing the objective based on quality improvements. □

D Prompt Templates

Translation Prompt Template

Translate the following text into {tgt_lang} without additional explanations:
{*src*}

Post-editing Prompt Template

Given the source text:
{*src*}
Improve the following draft {tgt_lang} translation into a high-quality {tgt_lang} version, without explanations:
{*pred*$_i$}

E Extended Results

E.1 Main Experiment

E.1.1 Evaluation

Large Models. For large-scale models such as Gemini-2.0-flash, DeepSeek-V3.2-Exp, and OpenAI GPT-5.2, we use the official APIs for evaluation. Only the prompt templates from Appendix D are used, with the maximum output length set to 512 tokens. All other generation parameters are left at their default settings.

Small Models. For smaller models, if an official translation prompt is available (e.g., Seed-X-PPO-7B, TowerInstruct-13B-v0.1), we use it; otherwise, we fall back to the prompt templates in Appendix D. During evaluation, sampling parameters are set to recommended defaults, as summarized in Table 5.

Model	Temp	Top-p	Top-k	Rep Pen
Seed-X-PPO-7B	0.0	-	-	-
TowerInstruct-13B-v0.1	0.0	-	-	-
Qwen3	0.6	0.95	20	1.05
MT-R1-Zero	0.6	0.95	20	1.05
Ours	0.6	0.95	20	1.05

Table 5: Sampling parameters for small models in translation experiments.

E.1.2 Training Dynamics

As RL training exhibits non-monotonic convergence, we report the performance trajectories underlying the main experimental results. Each training step processes 128 samples, and models are trained for 400 steps in total. Evaluation is performed on the **test set** every 20 steps, and the corresponding metrics are plotted to illustrate training dynamics over time, as shown in Figures 5–8.

E.2 Gradient Weight Analysis

We report the metric values at step 100 for the three experimental settings (Figure 3) in a table for clarity (Table 7).

E.3 Platform Generalization

Building upon verl, we extend pytorch_lightning, which is used by COMET models, with NPU support to enable the entire pipeline to run on NPU servers. We conduct experiments on a server equipped with 8× Ascend 910 NPUs.

Under comparable training budgets, we evaluate the performance against models trained on NVIDIA H20 (96GB) GPUs, as shown in Table 6.

Model	EN–FI (WMT24)		EN–FI (FLORES)		EN–TR (WMT24)		EN–TR (FLORES)	
	chrF++	Kiwi	chrF++	Kiwi	chrF++	Kiwi	chrF++	Kiwi
Ours-4B (Nvidia)	45.29	62.49	42.65	73.78	45.39	65.49	47.77	76.35
Ours-4B (Ascend)	45.15	63.07	42.32	73.32	45.06	65.41	47.29	76.09

Table 6: Comparison of Ours-4B trained on different hardware platforms. We report chrF++ and Kiwi on EN–FI and EN–TR translation tasks. Results show that the performance remains consistent across NVIDIA and Ascend platforms under comparable training settings.

The results demonstrate that our approach maintains consistent performance across different hardware platforms.

We release the weights of our 4B model at https://huggingface.co/collections/DGME/pegrl. The NVIDIA-based model is re-trained on A6000 GPUs, while the Ascend-based model is trained on the aforementioned NPU server.

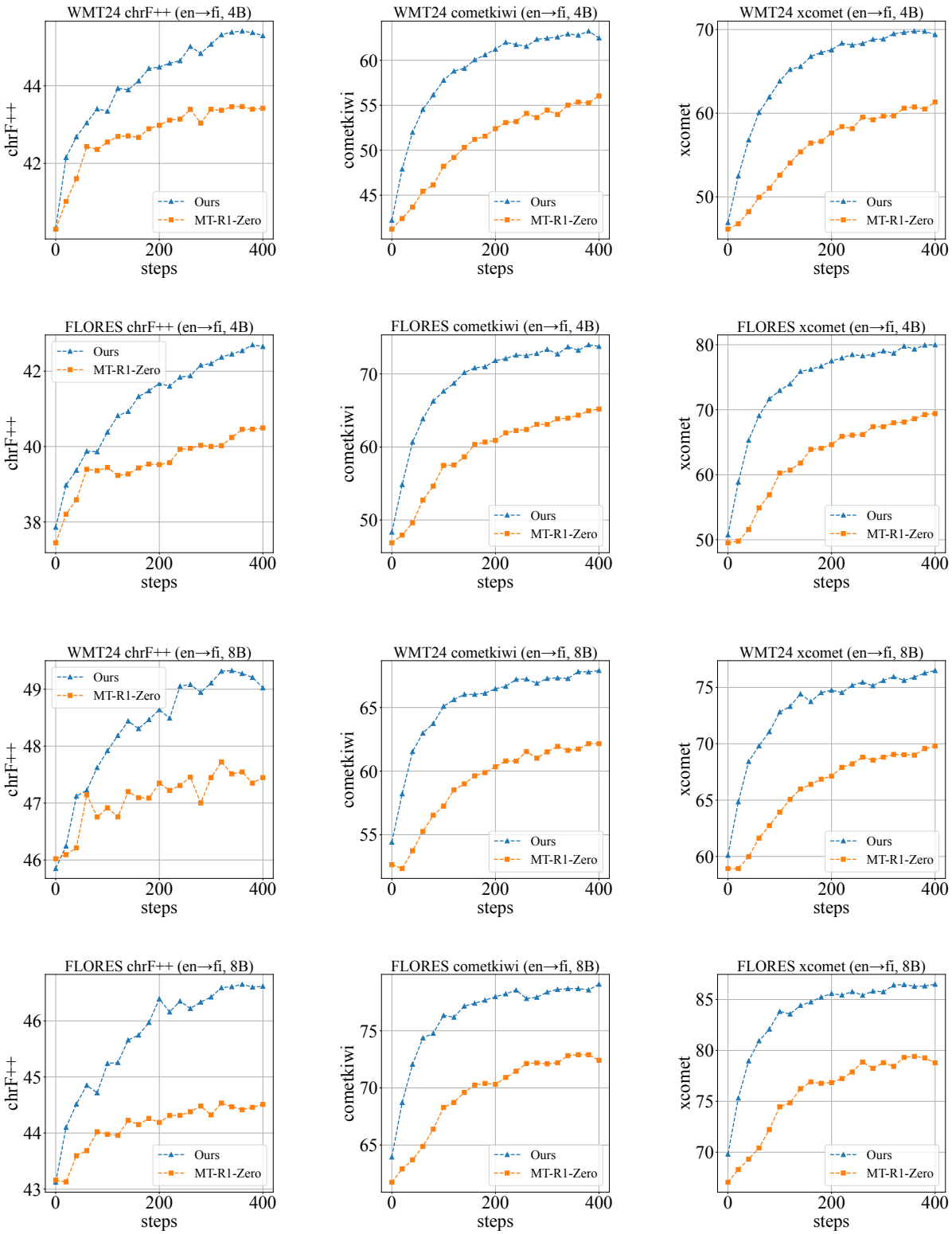

Figure 5: Training dynamics on FLORES and WMT24 for EN→FI under different model scales (4B, 8B), evaluated by chrF++, COMET-Kiwi, and XCOMET.

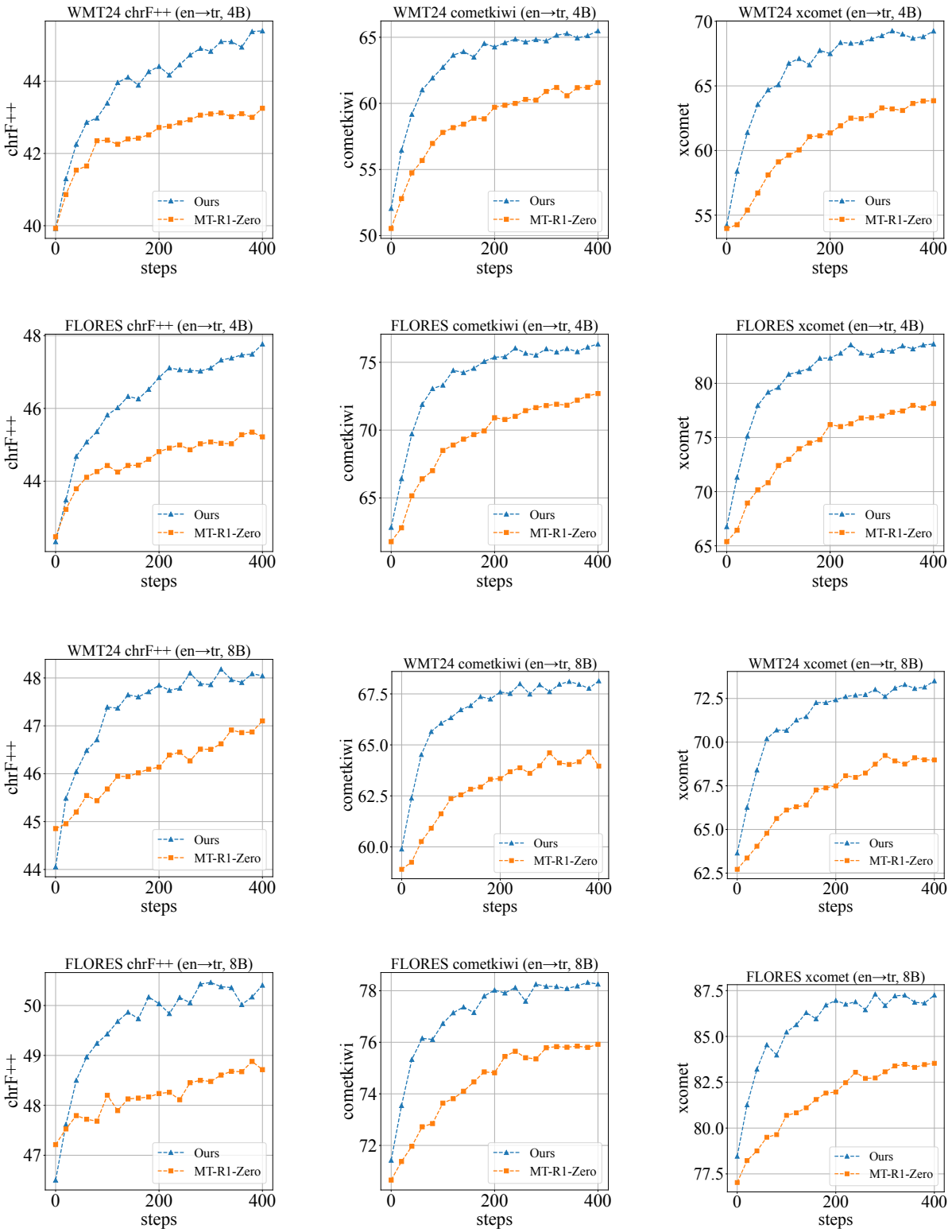

Figure 6: Training dynamics on FLORES and WMT24 for EN→TR under different model scales (4B, 8B), evaluated by chrF++, COMET-Kiwi, and XCOMET.

λ_{pe}	λ_{mt}	chrF++	COMETKIWI
M	1	43.79	56.96
M	0	42.48 (↓ 1.31)	55.80 (↓ 1.16)
1	1	43.06 (↓ 0.73)	53.09 (↓ 3.87)

Table 7: Performance at step 100 (corresponding to Figure 3). Values in subsequent rows are compared to the first row.

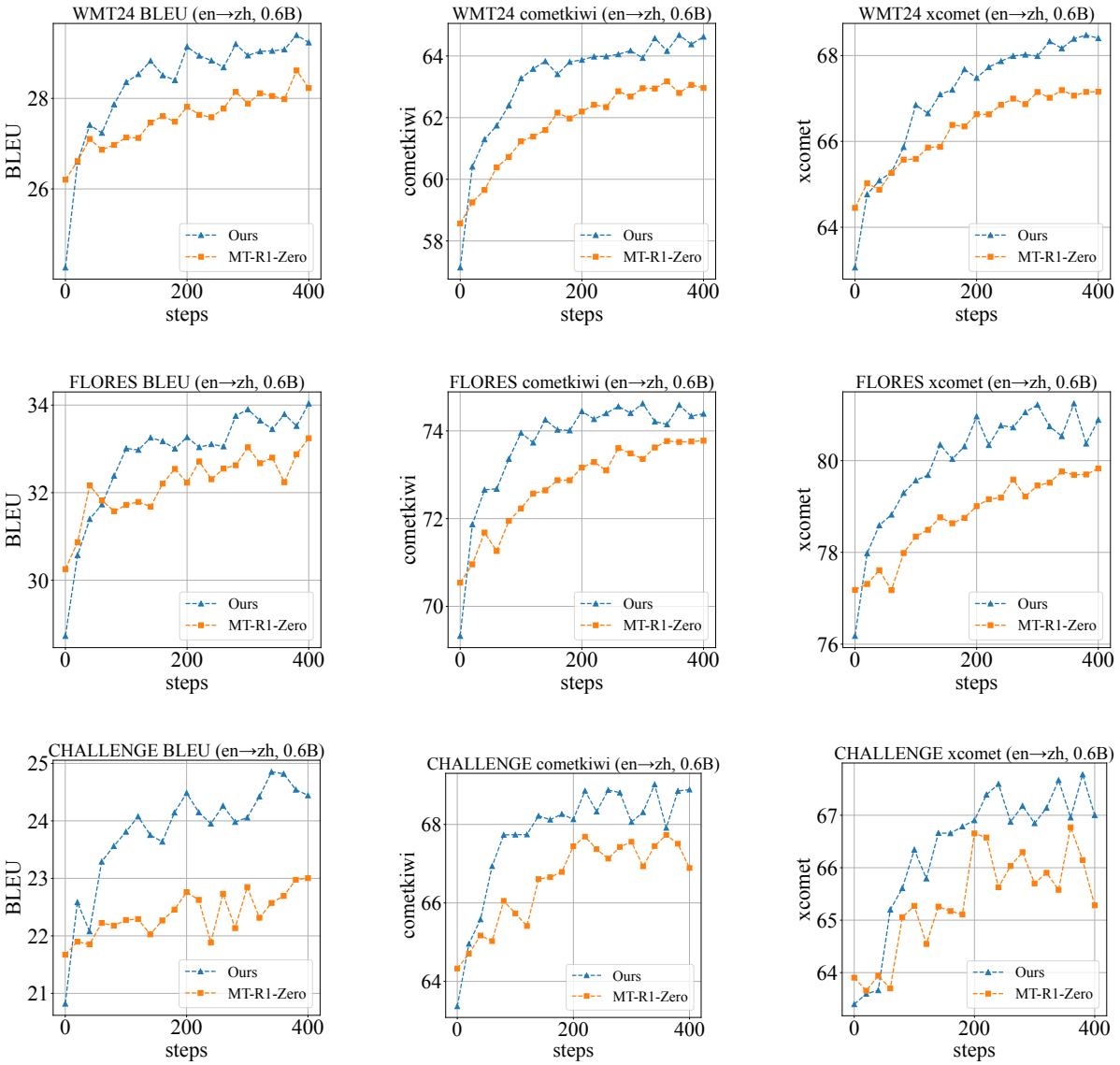

Figure 7: Training dynamics on FLORES, WMT24, and Challenge for EN→ZH with a 0.6B model, evaluated by BLEU, COMET-Kiwi, and XCOMET.

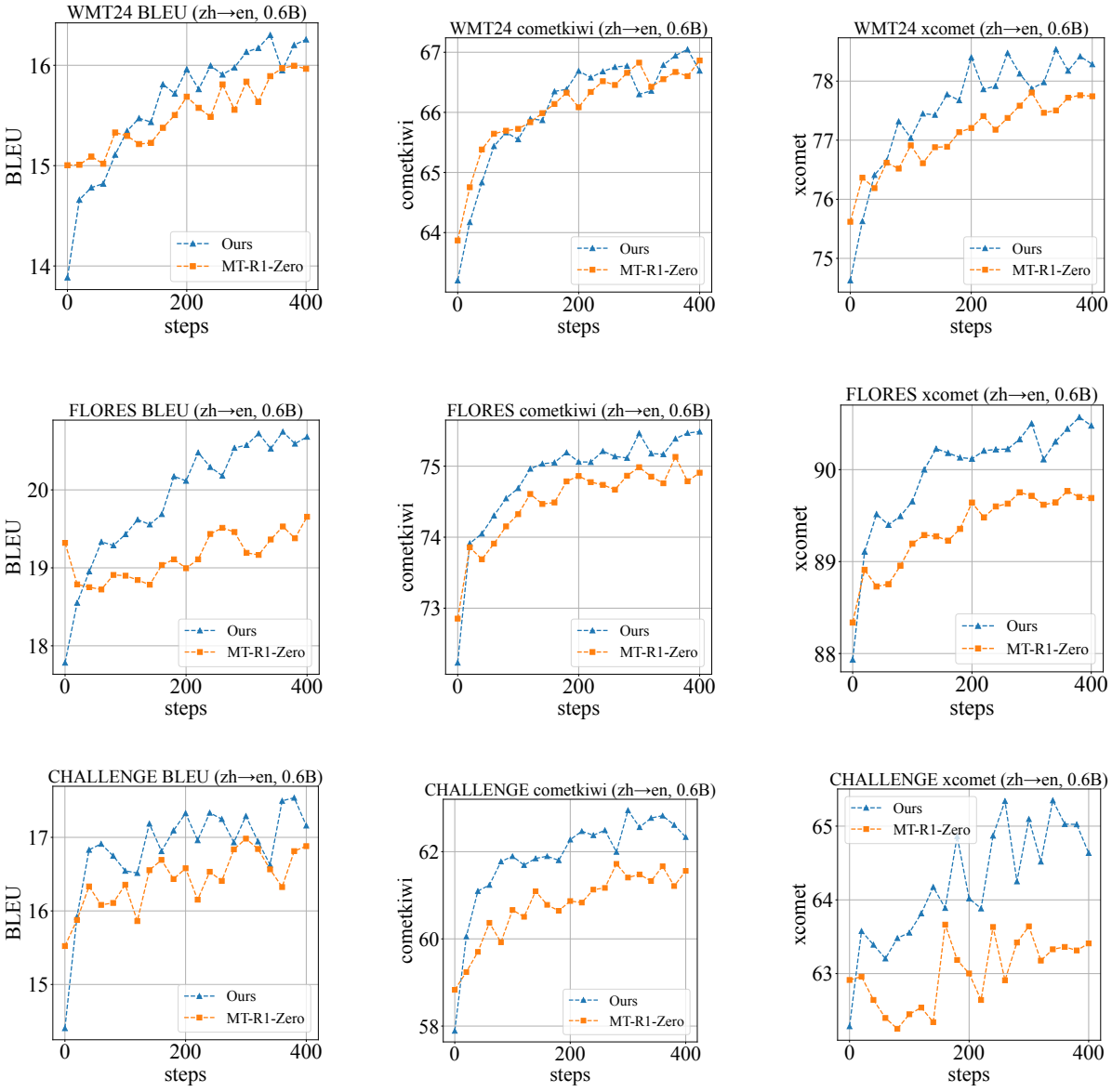

Figure 8: Training dynamics on FLORES, WMT24, and Challenge for ZH→EN with a 0.6B model, evaluated by BLEU, COMET-Kiwi, and XCOMET.